

A. AMENDMENTS TO CLAIMS

Please cancel Claims 10-13, 17, 19-22, 32-37, 41, 43 and 44, add new Claims 50-69 and amend the claims as indicated hereinafter.

1 1. (CURRENTLY AMENDED) A method for maintaining data integrity, comprising the
2 computer-implemented steps of:
3 generating checksum data by performing a physical checksum calculation on a block of
4 ~~data in volatile memory; data;~~
5 after generating said checksum data,
6 performing a logical check on data contained within the block of data; and
7 if the block of data passes said logical check, then causing the block of data to be
8 written to nonvolatile memory.

C\

1 2. (ORIGINAL) The method of Claim 1 wherein the steps of generating checksum data and
2 performing a logical check are performed in response to a request to write said block of
3 data to nonvolatile memory.

1 3. (ORIGINAL) The method of Claim 1 further comprising the step of writing the
2 checksum data to nonvolatile memory in association with writing said block of data to
3 nonvolatile memory.

1 4. (ORIGINAL) The method as recited in Claim 3, further comprising the steps of:
2 after writing the block of data to nonvolatile memory,
3 causing the block of data and said checksum data to be read from nonvolatile
4 memory; and

5 performing a physical checksum verification procedure on said block of data
6 based on said checksum data, wherein the physical checksum verification
7 procedure indicates whether the block of data was corrupted subsequent to
8 performing the logical check on the data contained with the block of data.

1 5. (ORIGINAL) The method as recited in Claim 1, further comprising the step of
2 performing one or more physical checksum verification procedures prior to writing the
3 block of data to nonvolatile memory, wherein the one or more physical checksum
4 verification procedures indicate whether the block of data was corrupted subsequent to
5 generating said checksum data.

1 6. (ORIGINAL) The method as recited in Claim 1, wherein:
2 the step of performing a physical checksum calculation comprises the step of a software
3 application performing the physical checksum calculation on said block of data;
4 and
5 the step of performing a logical check on data contained with the block of data comprises
6 the step of said software application performing the logical check on data
7 contained with the block of data.

1 7. (ORIGINAL) The method as recited in Claim 4, wherein:
2 the step of performing a physical checksum calculation comprises the step of a software
3 application performing the physical checksum calculation on said block of data;
4 the step of performing a logical check on data contained with the block of data comprises
5 the step of said software application performing the logical check on data
6 contained with the block of data; and

the step of performing a physical checksum verification procedure on said block of data comprises the step of said software application performing the physical checksum verification procedure on said block of data.

(ORIGINAL) The method as recited in Claim 5, wherein:

the step of performing a physical checksum calculation comprises the step of an software

application performing the physical checksum calculation on said block of data;

the step of performing a logical check on data contained with the block of data comprises

the step of said software application performing the logical check on data

contained with the block of data; and

the step of performing one or more physical checksum verification procedures prior to

writing the block of data to nonvolatile memory comprises the step of one or more

components other than said software application performing the one or more

physical checksum verification procedures prior to writing the block of data to

nonvolatile memory.

(ORIGINAL) The method as recited in Claim 4, further comprising the step of:

after performing the physical checksum verification procedure on said block of data,

storing the block of data as a backup version of the block of data, wherein the

backup version of the block of data is maintained separate from said block of data.

in said nonvolatile memory.

1 10 - 13. (CANCELED)

1 14. (CURRENTLY AMENDED) A method for maintaining data integrity, comprising the
2 computer-implemented steps of:
3 a software application performing a physical checksum calculation on a block of data;
4 after performing the physical checksum calculation,
5 a component other than the software application performing a first physical
6 checksum verification procedure on said block of data prior to writing the
7 block of data to ~~nonvolatile~~ a nonvolatile memory, wherein the first
8 physical checksum verification procedure indicates whether the block of
9 data was corrupted subsequent to the software application performing the
10 physical checksum calculation on the ~~data contained with the~~ block of
11 data; and
12 if the block of data passes said first physical checksum verification procedure,
13 then causing the block of data to be written to ~~nonvolatile~~ the nonvolatile
14 memory.

1 15. (CURRENTLY AMENDED) The method as recited in Claim 14, further comprising the
2 steps of:
3 after writing the block of data to ~~nonvolatile~~ the nonvolatile memory,
4 causing the block of data to be read from ~~nonvolatile~~ the nonvolatile memory; and
5 performing a second physical checksum verification procedure on said block of
6 data, wherein the second physical checksum verification procedure
7 indicates whether the block of data was corrupted subsequent to
8 performing the first physical checksum verification procedure on the ~~data~~
9 ~~contained with the~~ block of data.

1 16. (CURRENTLY AMENDED) The method as recited in Claim 14, further comprising
2 ~~wherein the step of performing a first physical checksum verification procedure includes~~
3 ~~the steps of performing a plurality of one or more other physical checksum verification~~
4 ~~procedures on said block of data prior to writing the block of data to nonvolatile the~~
5 ~~nonvolatile memory, wherein the plurality of one or more other physical checksum~~
6 ~~verification procedures indicate whether the block of data was corrupted subsequent to~~
7 ~~performing the physical checksum calculation on the data contained with the block of~~
8 ~~data.~~

1 17. (CANCELED)

1 18. (CURRENTLY AMENDED) The method as recited in ~~Claim 17, Claim 14~~, wherein the
2 step of a component other than the software application performing a first physical
3 checksum verification procedure on said block of data comprises the step of a disk array
4 component performing the first physical checksum verification procedure on said block
5 of data, wherein the disk array component is configured to write the block of data to disk
6 only after verifying the integrity of the data block.

1 19 - 22. (CANCELED).

1 23. (CURRENTLY AMENDED) A computer-readable medium for maintaining data
2 integrity, the computer-readable medium carrying one or more sequences of one or more
3 instructions which, when executed by one or more processors, cause the one or more
4 processors to perform the steps of:

5 generating checksum data by performing a physical checksum calculation on a block of
6 ~~data in volatile memory;~~ data;
7 after generating said checksum data,
8 performing a logical check on data contained within the block of data; and
9 if the block of data passes said logical check, then causing the block of data to be
10 written to nonvolatile memory.

1 24. (PREVIOUSLY PRESENTED) The computer-readable medium of Claim 23 wherein the
2 steps of generating checksum data and performing a logical check are performed in
3 response to a request to write said block of data to nonvolatile memory.

1 C) 25. (PREVIOUSLY PRESENTED) The computer-readable medium of Claim 23 further
2 comprising one or more sequences of additional instructions which, when executed by
3 the one or more processors, cause the one or more processors to perform the step of
4 writing the checksum data to nonvolatile memory in association with writing said block
5 of data to nonvolatile memory.

1 26. (PREVIOUSLY PRESENTED) The computer-readable medium as recited in Claim 25,
2 further comprising one or more sequences of additional instructions which, when
3 executed by the one or more processors, cause the one or more processors to perform the
4 steps of:
5 after writing the block of data to nonvolatile memory,
6 causing the block of data and said checksum data to be read from nonvolatile
7 memory; and

8 performing a physical checksum verification procedure on said block of data
9 based on said checksum data, wherein the physical checksum verification
10 procedure indicates whether the block of data was corrupted subsequent to
11 performing the logical check on the data contained with the block of data.

1 27. (PREVIOUSLY PRESENTED) The computer-readable medium as recited in Claim 23,
2 further comprising one or more sequences of additional instructions which, when
3 executed by the one or more processors, cause the one or more processors to perform the
4 step of performing one or more physical checksum verification procedures prior to
5 writing the block of data to nonvolatile memory, wherein the one or more physical
6 checksum verification procedures indicate whether the block of data was corrupted
7 subsequent to generating said checksum data.

1 28. (PREVIOUSLY PRESENTED) The computer-readable medium as recited in Claim 23,
2 wherein:
3 the step of performing a physical checksum calculation comprises the step of a software
4 application performing the physical checksum calculation on said block of data;
5 and
6 the step of performing a logical check on data contained with the block of data comprises
7 the step of said software application performing the logical check on data
8 contained with the block of data.

1 29. (PREVIOUSLY PRESENTED) The computer-readable medium as recited in Claim 26,
2 wherein:

3 the step of performing a physical checksum calculation comprises the step of an software
4 application performing the physical checksum calculation on said block of data;
5 the step of performing a logical check on data contained with the block of data comprises
6 the step of said software application performing the logical check on data
7 contained with the block of data; and
8 the step of performing a physical checksum verification procedure on said block of data
9 comprises the step of said software application performing the physical checksum
10 verification procedure on said block of data.

1 30. (PREVIOUSLY PRESENTED) The computer-readable medium as recited in Claim 27,
2 wherein:

3 the step of performing a physical checksum calculation comprises the step of an software
4 application performing the physical checksum calculation on said block of data;
5 the step of performing a logical check on data contained with the block of data comprises
6 the step of said software application performing the logical check on data
7 contained with the block of data; and
8 the step of performing one or more physical checksum verification procedures prior to
9 writing the block of data to nonvolatile memory comprises the step of one or more
10 components other than said software application performing the one or more
11 physical checksum verification procedures prior to writing the block of data to
12 nonvolatile memory.

1 31. (PREVIOUSLY PRESENTED) The computer-readable medium as recited in Claim 26,
2 further comprising one or more sequences of additional instructions which, when

executed by the one or more processors, cause the one or more processors to perform the step of:

after performing the physical checksum verification procedure on said block of data,

storing the block of data as a backup version of the block of data, wherein the

backup version of the block of data is maintained separate from said block of data

in said nonvolatile memory.

32 - 37. (CANCELED)

38. (CURRENTLY AMENDED) A computer-readable medium for storing data in a nonvolatile memory, the computer-readable medium carrying one or more sequences of one or more instructions which, when executed by one or more processors, cause the one or more processors to perform the steps of:

a software application performing a physical checksum calculation on a block of data;
after performing the physical checksum calculation,

a component other than the software application performing a first physical checksum verification procedure on said block of data prior to writing the block of data to ~~nonvolatile~~ a nonvolatile memory, wherein the first physical checksum verification procedure indicates whether the block of data was corrupted subsequent to the software application performing the physical checksum calculation on the ~~data contained with the~~ block of data; and

if the block of data passes said first physical checksum verification procedure,

then causing the block of data to be written to ~~nonvolatile~~ the nonvolatile memory.

1 39. (CURRENTLY AMENDED) The computer-readable medium as recited in Claim 38,
2 further comprising one or more sequences of additional instructions which, when
3 executed by the one or more processors, cause the one or more processors to perform the
4 steps of:
5 after writing the block of data to ~~nonvolatile~~ the nonvolatile memory,
6 causing the block of data to be read from ~~nonvolatile~~ the nonvolatile memory; and
7 performing a second physical checksum verification procedure on said block of
8 data, wherein the second physical checksum verification procedure
9 indicates whether the block of data was corrupted subsequent to
10 performing the first physical checksum verification procedure on the ~~data~~
11 ~~contained with the~~ block of data.

1 40. (CURRENTLY AMENDED) The computer-readable medium as recited in Claim 38,
2 further comprising wherein the step of performing a first physical checksum verification
3 procedure includes the steps of performing a plurality of one or more other physical
4 checksum verification procedures on said block of data prior to writing the block of data
5 to nonvolatile the nonvolatile memory, wherein the plurality of one or more other
6 physical checksum verification procedures indicate whether the block of data was
7 corrupted subsequent to performing the physical checksum calculation on the data
8 contained with the block of data.

1 41. (CANCELED)

1 42. (CURRENTLY AMENDED) The computer-readable medium as recited in Claim 41,
2 Claim 38, wherein the step of a component other than the software application

3 performing a first physical checksum verification procedure on said block of data
4 comprises the step of a disk array component performing the first physical checksum
5 verification procedure on said block of data, wherein the disk array component is
6 configured to write the block of data to disk only after verifying the integrity of the data
7 block.

1 43 - 44. (CANCELED)

1 45. (CURRENTLY AMENDED) A storage device comprising:
2 a software application configured to perform a physical checksum calculation on a block
3 of data; and
4 a component configured to
5 perform a first physical checksum verification procedure on the block of data prior
6 to writing the block of data to a nonvolatile memory, wherein the first
7 physical checksum verification procedure indicates whether the block of
8 data was corrupted subsequent to the software application performing the
9 physical checksum calculation on the block of data, and
10 if the block of data passes said first physical checksum verification procedure,
11 then cause the block of data to be written to the nonvolatile memory.
12 ~~a storage medium; and~~
13 ~~a storage mechanism communicatively coupled to the storage medium, the storage~~
14 ~~mechanism being configured to:~~
15 ~~perform a set of one or more verifications on a block of data prior to~~
16 ~~allowing the block of data to be written to the storage medium,~~
17 ~~wherein the set of one or more verifications includes a physical~~

18 checksum verification that indicates whether the block of data was
19 corrupted subsequent to a physical checksum calculation being
20 performed on the data contained with the block of data, and
21 if the block of data passes the set of verifications, then allow the block of
22 data to be written to the storage medium.

1 46. (CURRENTLY AMENDED) The storage device as recited in Claim 45, wherein the
2 component is further configured to after writing the block of data to the nonvolatile
3 memory.

4 cause the block of data to be read from the nonvolatile memory; and
5 perform a second physical checksum verification procedure on said block of data,
6 wherein the second physical checksum verification procedure indicates
7 whether the block of data was corrupted subsequent to the component
8 performing the first physical checksum verification procedure on the block
9 of data.

10 set of verifications includes a logical check on the data contained within the block of data.

1 47. (CURRENTLY AMENDED) The storage device as recited in Claim 46, wherein the
2 ~~storage mechanism component~~ is further configured to perform ~~one or more other~~
3 physical checksum verification procedures on said block of data prior to writing the block
4 of data to ~~nonvolatile~~ the nonvolatile memory, wherein the ~~plurality of~~ one or more other
5 physical checksum verification procedures indicate whether the block of data was
6 corrupted subsequent to performing the physical checksum calculation on the ~~data~~

7 contained with the block of data. the logical check after the physical checksum
8 verification.

1 48. (CURRENTLY AMENDED) The storage device as recited in Claim 45, wherein the
2 component is a disk array. storage mechanism is further configured to:
3 cause the block of data to be read from the storage medium, and
4 perform a second physical checksum verification on the block of data that indicates
5 whether the block of data was corrupted subsequent to the prior physical
6 checksum verification being performed on the block of data.

1 49. (PREVIOUSLY PRESENTED) A storage device comprising:
2 a storage medium; and
3 a storage mechanism communicatively coupled to the storage medium, the storage
4 mechanism being configured to:
5 perform a logical check on data contained in a block of data after a physical
6 checksum calculation has previously been performed on the block of data,
7 and
8 if the block of data passes the logical check, then allowing the block of data to be
9 written to the storage medium.

1 50. (NEW) The apparatus as recited in Claim 49, wherein the physical checksum calculation
2 and the logical check are both performed in response to a request to write the block of
3 data to the storage medium.

1 51. (NEW) The apparatus as recited in Claim 49, wherein the storage mechanism is further
2 configured to write, to the storage medium, checksum data generated by the physical
3 checksum calculation.

1 52. (NEW) The apparatus as recited in Claim 51, wherein the storage mechanism is further
2 configured to
3 after the block of data is written to the storage medium,
4 cause the block of data and the checksum data to be read from the storage
5 medium; and
6 perform a physical checksum verification on the block of data based on the
7 checksum data, wherein the physical checksum verification indicates
8 whether the block of data was corrupted subsequent to performing the
9 logical check on the data contained with the block of data.

1 C
2 53. (NEW) The apparatus as recited in Claim 49, wherein the storage mechanism is further
3 configured to perform one or more physical checksum verifications prior to allowing the
4 block of data to be written to the storage medium, wherein the one or more physical
5 checksum verifications indicate whether the block of data was corrupted subsequent to
the physical checksum calculation.

1 54. (NEW) The apparatus as recited in Claim 49, wherein both the physical checksum
2 calculation and the logical check are performed by a software application.

1 55. (NEW) The apparatus as recited in Claim 52, wherein the physical checksum calculation,
2 the physical checksum verification and the logical check are performed by a software
3 application.

1 56. (NEW) The apparatus as recited in Claim 53, wherein both the physical checksum
2 calculation and the logical check are performed by a software application and the
3 physical checksum verification is performed by one or more components other than the
4 software application.

1 57. (NEW) The apparatus as recited in Claim 52, wherein the storage
2 mechanism is further configured to after performing the physical checksum
3 verification on the block of data, causing the block of data to be stored on
4 the storage medium as a backup version of the block of data, wherein the
5 backup version of the block of data is maintained separate from the block of
6 data on the storage medium.

1 58. (NEW) A method for maintaining data integrity, the method comprising the computer-
2 implemented steps of:
3 performing a physical checksum calculation on a block of data;
4 performing a logical check on the block of data;
5 performing a physical checksum verification on the block of data, wherein the physical
6 checksum verification indicates whether the block of data was corrupted
7 subsequent to performing the physical checksum calculation on the block of data;
8 and

9 if the block of data passes both the logical check and the physical checksum verification,
10 then allowing the block of data to be written to a nonvolatile memory.

1 59. (NEW) The method as recited in Claim 58, wherein the logical check is performed prior
2 to the performance of the physical checksum verification.

1 60. (NEW) The method as recited in Claim 58, wherein:
2 a software application performs both the physical checksum calculation and the logical
3 check on the block of data;
4 one or more components other than the software application perform the physical
5 checksum verification.

1 61. (NEW) The method as recited in Claim 58, wherein the logical check is performed after
2 the physical checksum calculation.

1 62. (NEW) A computer-readable medium for maintaining data integrity, the computer-
2 readable medium carrying one or more sequences of instructions which, when executed
3 by one or more processors, cause performance of the steps of:
4 performing a physical checksum calculation on a block of data;
5 performing a logical check on the block of data;
6 performing a physical checksum verification on the block of data, wherein the physical
7 checksum verification indicates whether the block of data was corrupted
8 subsequent to performing the physical checksum calculation on the block of data;
9 and
10 if the block of data passes both the logical check and the physical checksum verification,
11 then allowing the block of data to be written to a nonvolatile memory.

1 63. (NEW) The computer-readable medium as recited in Claim 62, wherein the logical check
2 is performed prior to the performance of the physical checksum verification.

1 64. (NEW) The computer-readable medium as recited in Claim 62, wherein:
2 a software application performs both the physical checksum calculation and the logical
3 check on the block of data;
4 one or more components other than the software application perform the physical
5 checksum verification.

1 65. (NEW) The computer-readable medium as recited in Claim 62, wherein the logical check
2 is performed after the physical checksum calculation.

C
1 66. (NEW) An apparatus for maintaining data integrity, the apparatus comprising a memory
2 storing instructions which, when executed by one or more processors, cause performance
3 of the steps of:
4 performing a physical checksum calculation on a block of data;
5 performing a logical check on the block of data;
6 performing a physical checksum verification on the block of data, wherein the physical
7 checksum verification indicates whether the block of data was corrupted
8 subsequent to performing the physical checksum calculation on the block of data;
9 and
10 if the block of data passes both the logical check and the physical checksum verification,
11 then allowing the block of data to be written to a nonvolatile memory.

1 67. (NEW) The apparatus as recited in Claim 66, wherein the logical check is performed
2 prior to the performance of the physical checksum verification.

1 68. (NEW) The apparatus as recited in Claim 66, wherein:
2 a software application performs both the physical checksum calculation and the logical
3 check on the block of data;
4 one or more components other than the software application perform the physical
5 checksum verification.

1 69. (NEW) The apparatus as recited in Claim 66, wherein the logical check is performed
2 after the physical checksum calculation.
